

# LAB ASSIGNMENT: Digital Signature and PKI. OpenSSL

---

CRYPTOGRAPHY AND COMPUTER SECURITY

Ana I. González-Tablas Ferreres  
José María de Fuentes García-Romero de Tejada  
Lorena González Manzano  
Sergio Pastrana Portillo  
UC3M | GRUPO COMPUTER SECURITY LAB (COSEC)



## TOOLS

OpenSSL is a cryptographic framework available in Linux distributions, like Ubuntu. Pay attention to the changes taking place while executing the commands proposed in this assignment.

OPENSSL. Available in most Linux by default. For Windows, follow specifications: <https://wiki.openssl.org/index.php/Binaries>

- For Windows systems, in order to execute the command from anywhere in the system, you must include the *bin* folder of OpenSSL in the environment variable PATH. Use the following command:

```
set PATH=%PATH%; [OPENSSL_INSTALL_PATH]/bin
```

## INTRODUCTION

This lab is structured in three different parts: 1) Creation of a PKI, 2) Digital Signature, 3) Obtaining a certificate through a website

The objective of this practice is to understand the concepts underlying a public key infrastructure based on the hierarchical trust model.

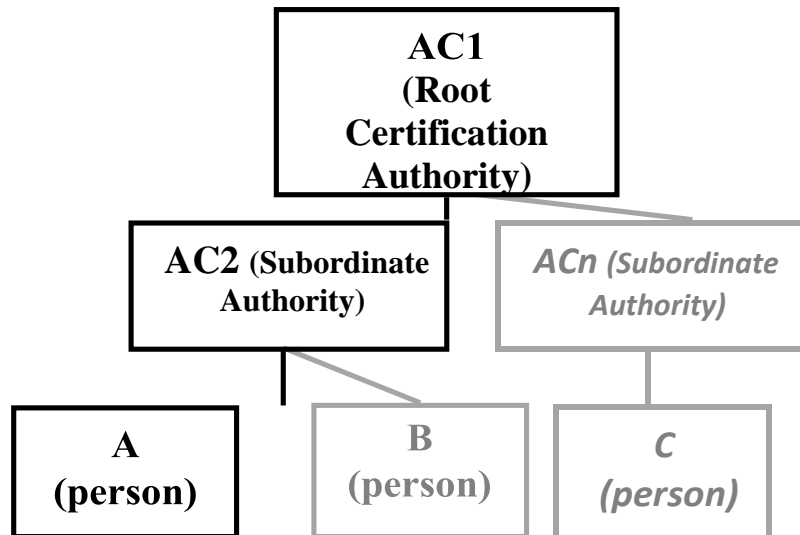
Specifically, the objectives are the following:

1. Understand required steps in order to an Authority issues a certificate
2. Understand causes that produce a certificate revocation and steps required to carry out this process.
3. Understand what the certificates role is regarding the signing and verifying documents.

To achieve these objectives each group becomes a ROOT CERTIFICATION AUTHORITY (equal to the “Fábrica Nacional de Moneda y Timbre” is in the real world). Such Authority (AC1), according to organizational reasons (for example, to

have a local office in all different districts), has some SUBORDINATE CERTIFICATION AUTHORITIES (AC2, AC3,...CAn). Moreover, these last Authorities are in charge of issuing public key certificates to people (A,B,C...)

The group of all the Authorities composes a Public Key Infrastructure (PKI).



To reduce the amount of work, in this assignment only the root certification authority (AC1), a single subordinate authority (AC2) and a person (A) must be configured.

To help with the organization of the practice, create 3 directories, one for each entity: AC1, AC2 and A .

```
# lab> mkdir AC1 AC2 A
```

**In order to issue the certificates, the Authorities use a certification POLICY. Copy the configuration files `openssl_AC1.cnf` and `openssl_AC2.cnf` (available in Aula Global) to the corresponding directories AC1 and AC2.** Analyze these files against the default configuration file (in Linux: `/etc/ssl/openssl.cnf`)

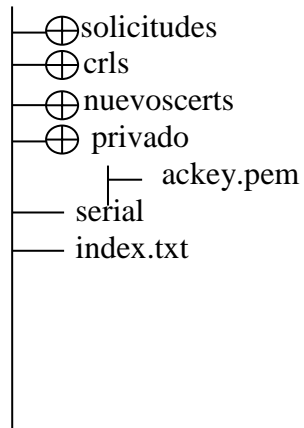
Before starting the assignment, change de configuration file such that the name of each AC will be AC1 XXXXX and AC2 XXXXX, being XXXXX the five last digits of the student ID.

Relevant OpenSSL commands to do the assignment:

- ca: enables creating and managing a Certification Authority according to the hierarchical model.
- req: enables creating and managing certificate request
- x509: enables managing X.509 certificates
- verify: enables verifying X.509 certificates

Each AC directory should have the following structure:

AC



## ***Configuration of AC1 (Root CA)***

1. Generate the directory structure necessary for AC1 and initialize the files serial and index.txt.

```
# AC1> mkdir solicitudes crls nuevoscerts
privado # AC1> echo '01' > serial
# AC1> touch index.txt
```

2. Generate the RSA key pair and the self signed certificate for AC1. Analyze the output.

```
# AC1> openssl req -x509 -newkey rsa:2018 -days 360 -out
ac1cert.pem -outform PEM -config openssl_AC1.cnf
# AC1> openssl x509 -in ac1cert.pem -text -noout
```

## ***Configuration of AC2 (subordinate CA)***

3. Generate the directory structure necessary for AC2 and initialize the files serial and index.txt.

```
# AC2> mkdir solicitudes crls nuevoscerts
privado # AC2> echo '01' > serial
# AC2> touch index.txt
```

4. Generate the RSA key pair for AC2 and the certificate request which will be sent to AC1 and 'send' it to AC1. Analyze the results.

```
# AC2> openssl req -newkey rsa:2048 -days 360 -out
ac2req.pem - outform PEM -config openssl_AC2.cnf
```

As it happened in AC1, a passphrase will be required. This has to be remembered for each operation that involves AC2's private key

```
# AC2> openssl req -in ac2req.pem -text
-noout # AC2> cp ac2req.pem
../AC1/solicitudes
```

## ***Generation of AC2's certificate by AC1***

5. Verify the request "sent" by AC2.

```
# AC1> openssl req -in ./solicitudes/ac2req.pem -text -noout
```

6. Generate the corresponding certificate for AC2 and 'send' it back to AC2. Rename 01.pem into ac2cert.pem, because AC2 as this same in its configuration file.

```
# AC1> openssl ca -in ./solicitudes/ac2req.pem -  
notext - extensions v3_subca -config  
openssl_AC1.cnf  
  
# AC1> cp ./nuevoscerts/01.pem ../AC2/ac2cert.pem
```

### ***Generation of keys for entity A as well as its corresponding certificate request***

7. For entity A, generate an RSA key pair as well as a certificate request and “send” it to AC2 (when generating the certificate requests, fill in ALL the requested fields and indicate “ES” as country, “MADRID” as province, “UC3M” as organization, and common name is XXXXX (as described before) and your email is your student email.

```
# A> openssl req -newkey rsa:1024 -days 360 -sha1  
-keyout Akey.pem -out Areq.pem  
  
# A> openssl req -in Areq.pem -text  
-noout # A> cp Areq.pem  
../AC2/solicitudes
```

### ***Generation of A certificate by AC2***

8. Verify the request “sent” by A.

```
# AC2> openssl req -in ./solicitudes/Areq.pem -text -noout
```

9. Generate certificate for A and “send” it back to this entity.

```
# AC2> openssl ca -in ./solicitudes/Areq.pem -notext -config  
./openssl_AC2.cnf  
  
# AC2> cp ./nuevoscerts/01.pem ../A/Acert.pem
```

10. Analyze changes in AC2 directory and check the resulting certificate:

```
# A> openssl x509 -in Acert.pem -text -noout
```

## ***Verification of A certificate***

11. Obtain a copy of the public key certificates of AC1 and AC2 and verify (you need to concatenate both AC1 and AC2 certificates in a single file).

```
# A> cp
../AC1/ac1cert.pem ./
# A> cp
../AC2/ac2cert.pem ./
# A> cat ac1cert.pem ac2cert.pem > certs.pem
# A> openssl verify -CAfile certs.pem Acert.pem
```

## ***Joining the certificate and the private key to sign in common applications (Word/ Email)***

12. Export the certificate of entity A, its private key and both AC1 and AC2 certificates (file certs.pem) to PKCS12 format.

```
# A> openssl pkcs12 -export -in Acert.pem -inkey
Akey.pem -certfile certs.pem -out Acert.p12
```

NOTE: First, A's passphrase is requested to export the private key, and then a new passphrase to protect the .p12 certificate

## **EJERCICIOS**

### **Exercise 1 :**

- What is the purpose of "serial" and "index.txt" files?
- What is the purpose of the "index.txt" file?
- Can AC2 create its certificate using step 2?
- If you were a real world Certification Authority, give a reasoned explanation (i.e. pros and cons, other possible choices, etc.) for each one of the following parameters of your certification policy:

## **Using A's private key to sign a document**

13. Create a document in Microsoft Word and sign it digitally using A's private key. First, you need to import the file Acert.p12 into your browser and then, using Microsoft Word, go to the option Office > Prepare > Add a digital signature ...

### **Exercise 2:**

- e) Regarding the document Word just created and signed, you will see an "Verification Error" message. What is the cause? Could you solve this?

## **Web Browser**

- f) What steps does a citizen carry out to obtain a public key certificate (e.g. from Spanish FNMT)? Fill the following table associating each step carried out to obtain a certificate through the FNMT with each step carried out to obtain a certificate in this assignment.

| <b>Step</b> | <b>Description</b>                     | <b>Step of this script where it has been performed</b> |
|-------------|--|--|
| <b>1</b>    | Create a key pair (public and private) |  |
| <b>2</b>    | Go to a registration office            |  |
| <b>3</b>    | Create public key certificate          |  |
| <b>4</b>    | Download certificate                   |  |

Now going to the browser, let's see how X.509 certificates are used when surfing the Internet.

1. Open a browser and point it to aulaglobal.uc3m.es and [www.google.es](http://www.google.es)
  - a. Which is the certificate sent by each page?
  - b. Which is the certification path?
2. Repeat the process with the uc3m web page ([www.uc3m.es](http://www.uc3m.es))



## ***Web browser. Server side***

### **Exercise 3:**

14. Open a browser and point it to <https://letsencrypt.org>
  - a. Which is the purpose of this webpage?
  - b. Which are the pros of the server public key certificates issued by Let's encrypt?